Word Segmentation for Urdu OCR System

Misbah Akram
National University of Computer and Emerging Sciences
misbahakram@gmail.com

Sarmad Hussain
Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
sarmad.hussain@kics.edu.pk

Abstract

This paper presents a technique for word segmentation for the Urdu OCR system. Word segmentation or word tokenization is a preliminary task for Urdu language processing. Several techniques are available for word segmentation in other languages. A methodology is proposed for word segmentation in this paper which determines the boundaries of words given a sequence of ligatures, based on collocation of ligatures and words in the corpus. Using this technique, word identification rate of 96.10% is achieved, using trigram probabilities normalized over the number of ligatures and words in the sequence.

1 Introduction

Urdu uses Nastalique style of Arabic script for writing, which is cursive in nature. Characters join together to form ligatures, which end either with a space or with a non-joining character. A word may be composed of one of more ligatures. In Urdu, space is not used to separate two consecutive words in a sentence; instead readers themselves identify the boundaries of words, as the sequence of ligatures, as they read along the text. Space is used to get appropriate character shapes and thus it may even be used within a word to break the word into constituent ligatures (Naseem 2007, Durrani 2008). Therefore, like other languages (Theeramunkong & Usanavasin, 2001; Wan and Liu, 2007; Khankasikam & Muansuwan, 2005; Haruechaiyasak et al., 2008; Haizhou & Baosheng, 1998), word segmentation or word tokenization is a preliminary task for Urdu language processing. It has applications in many areas like spell checking, POS tagging, speech synthesis, information retrieval etc. This paper focuses on the word segmentation problem from the point of view of Optical Character Recognition (OCR) System. As space is not visible in typed and scanned text, spacing cues are not available to the OCR for word separation and therefore segmentation has to be done more explicitly. This word segmentation model for Urdu OCR system takes input in the form of a sequence of ligatures recognized by an OCR to construct a sequence of words from them.

2 Literature Review

Many languages, e.g., English, French, Hindi, Nepali, Sinhala, Bengali, Greek, Russian, etc. segment text into a sequence of words using delimiters such as space, comma and semi colon etc., but on the other hand many Asian languages like Urdu, Persian, Arabic, Chinese, Dzongkha, Lao and Thai have no explicit word boundaries. In such languages, words are segmented using more advanced techniques, which can be categorized into three methods:

(i) Dictionary/lexicon based approaches
(ii) Linguistic knowledge based approaches
(iii) Machine learning based approaches/statistical approaches (Haruechaiyasak et al., 2008)

Longest matching (Poowarawan, 1986; Richard Sproat, 1996) and maximum matching (Sproat et al., 1996; Haizhou & Baosheng, 1998) are examples of lexicon based approaches. These techniques segment text using the lexicon. Their

Proceedings of the 8th Workshop on Asian Language Resources, pages 88–94,
Beijing, China, 21-22 August 2010.

accuracy depends on the quality and size of the dictionary.

N-Grams (Chang et al., 1992; Li Haizhou et al., 1997; Richard Sproat, 1996; Dai & Lee, 1994; Aroonmanakun, 2002) and Maximum collocation (Aroonmanakun, 2002) are Linguistic knowledge based approaches, which also rely very much on the lexicon. These approaches select most likely segmentation from the set of possible segmentations using a probabilistic or cost-based scoring mechanism.

Word segmentation using decision trees (Sornlertlamvanich et al., 2000; Theeramunkong & Usanavasin, 2001) and similar other techniques fall in the third category of word segmentation techniques. These approaches use a corpus in which word boundaries are explicitly marked. These approaches do not require dictionaries. In these approaches ambiguity problems are handled by providing a sufficiently large set of training examples to enable accurate classification.

A knowledge based approach has been adopted for earlier work on Urdu word segmentation (Durrani 2007; also see Durrani and Hussain 2010). In this technique word segmentation of Urdu text is achieved by employing knowledge based on the Urdu linguistics and script. The initial segmentations are ranked using min-word, unigram and bigram techniques. It reports 95.8 % overall accuracy for word segmentation of Urdu text. Mukund et al. (2009) propose using character model along with linguistic rules and report 83% precision. Lehal (2009) proposes a two stage process, which first uses Urdu linguistic knowledge, and then uses statistical information of Urdu and Hindi (also using transliteration into Hindi) in the second stage for words not addressed in the first stage, reporting an accuracy of 98.57%.

These techniques use characters or words in the input, whereas an OCR outputs a series of ligatures. The current paper presents work done using statistical methods as an alternative, which works with ligatures as input.

3 Methodology

Current work uses the co-occurrence information of ligatures and words to construct a statistical model, based on manually cleaned and segmented training corpora. Ligature and word statistics are derived from these corpora. In the decoding phase, first all sequences of words are generated from input set of ligatures and ranking of these sequences is done based on lexical lookup. Top *k* sequences are selected for further processing, based on the number of valid words. Finally, the probability of each of the *k* sequences is calculated for the final decision. Details are described in the subsequent sections.

3.1 Data collection and preparation

An existing lexicon of 49630 unique words is used (derived from Ijaz et al. 2007). The corpus used for building ligature grams consists of half a million words. Of these, 300,000 words are taken from the Sports, Consumer Information and Culture/Entertainment domains of the 18 million word corpus (Ijaz et al. 2007), 100,000 words are obtained from Urdu-Nepali-English Parallel Corpus (available at www.PANL10n.net), and another 100,000 words are taken from a previously POS tagged corpus (Sajjad, 2007; tags of this corpus are removed before further processing). This corpus is manually cleaned for word segmentation errors, by adding missing spaces between words and replacing spaces with Zero Width Non-Joiner (ZWNJ) within words. For the computation of word grams, the 18 million word corpus of Urdu is used (Ijaz et al. 2007).

3.2 Count and probability calculations

Table 1 and Table 2 below give the counts for unigram, bigrams and trigram of the ligatures and the words derived from the corpora respectively.

Ligature Tokens	Ligature Unigram	Ligature Bigrams	Ligature Trigrams
1508078	10215	35202	65962

Table 1. Unigram, bigram and trigram counts of the ligature corpus

Word Tokens	Word Unigrams	Word Bigrams	Word Trigrams
17352476	157379	1120524	8143982

Table 2. Unigram, bigram and trigram counts of the word corpus

After deriving word unigrams, bigrams, and trigrams, the following cleaning of corpus is

performed. In the 18 million word corpus, certain words are combined due to missing space, but are separate words. Some of these words occur with very high frequency in the corpus. For example "ہوگا" (*ho ga*, "will be") exists as single word rather than two words due to missing space. To solve this space insertion problem, a list of about 700 words with frequency greater than 50 is obtained from the word unigrams. Each word of the list is manually reviewed and space is inserted, where required. Then these error words are removed from the word unigram and added to the word unigram frequency list as two or three individual words incrementing respective counts.

For the space insertion problem in word bigrams, each error word in joined-word list (700-word list) is checked. Where these error words occurs in a bigram word frequency list, for example "کیا ہوگا" (*kiya ho ga* "will have done") exists in the bigram list and contains "ہوگا" error word, then this bigram entry "کیا ہوگا" is removed from the bigram list and counts of "ہو گا" and "کیا ہو" are increased by the count of "کیا ہوگا". If these words do not exist in the word bigram list then they are added as a new bigrams with the count of "کیا ہوگا". Same procedure is performed for the word trigrams.

The second main issue is with word-affixes, which are sometimes separated by spaces from the words. Therefore, in calculations, these are treated as separate words and exist as bigram entries in the list rather than a unigram entry. For example "صحت مند" (*sehat+mand*, "healthy") exists as a bigram entry but in Urdu it is a single word. To cope with this problem, a list of word-affixes is used. If any entry of word bigram matches with an affix, then this word is combined by removing spurious space from it (and inserting ZWNJ, if required to maintain its glyph shape). Then this word is inserted in the unigram list with its original bigram count and unigram list updated accordingly. Same procedure is performed if a trigram word matches with an affix.

After cleaning, unigram, bigram and trigram counts for both words and ligatures are calculated. To avoid data sparseness One Count Smoothing (Chen & Goodman, 1996) is applied.

3.3 Word sequences generation from input

The input, in the form of sequence of ligatures is used to generate all possible words. These sequences are then ranked based on real words. For this purpose, a tree of these sequences is incrementally built. The first ligature is added as a root of tree, and at each level two to three additional nodes are added. For example the second level of the tree contains the following tree nodes.

- Current ligature forms a separate word, separated with space, from the sequence at its parent, $l_1\ l_2$
- Current ligature concatenates, without a space, with the sequence at its parent, $l_1 l_2$
- Current ligature concatenates, without a space, with the sequence at its parent but with an additional, l_1ZWNJl_2

For each node, at each level of the tree, a numeric value is assigned, which is the sum of squares of the number of ligatures in each word which is in the dictionary. If a word does not exist in dictionary then it does not contribute to the total sum. If a node-string has only one word and this word does not occur in the dictionary as a valid word then it is checked that this word may occur at the start of any dictionary entry. In this case numeric value is also assigned.

After assignment, nodes are ranked according to these values and best *k* (beam value) nodes are selected. These selected nodes are further ranked using statistical methods discussed below.

3.4 Best word segmentation selection

For selection of the most probable word segmentation sequence word and ligature models are used. For word probabilities the following is used.

$$P(W) = \quad \text{argmax}_{w_1^n \in S} P(w_1^n)$$

To reduce the complexity of computing, Markov assumption are taken to give bigram and trigram approximations (e.g., see Jurafsky & Martin 2006) as given below.

$$P(W) = \quad \text{argmax}_{w_1^n \in S} \prod_1^n P(w_i | w_{i-1})$$
$$P(W) = \quad \text{argmax}_{w_1^n \in S} (\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})$$

Similarly the ligature models are built by taking the assumption that sentences are made

up of sequences of ligatures rather than words and space is also a valid ligature. By taking the Markov bigram and trigram assumption for ligature grams we get the following.

$$P(L) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}))$$
$$P(L) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}l_{i-2}))$$

Given the ligatures, e.g. as input from and OCR, we can formulate the decoding problem as the following equation.

$$P(W|L) = \operatorname{argmax}_{w_1^n \in S} P(w_1^n|l_1^m)$$

where $w_1^n = w_1, w_2, w_3, w_4, \ldots w_n$ and $l_1^m = l_1, l_2, l_3, l_4, \ldots l_m$; n represents number of words and m represents the number of ligatures. This equation also represents that m number of ligatures can be assigned to n number of words. By applying the Bayesian theorem we get the following derivation.

$$P(W|L) = \operatorname{argmax}_{w_1^n \in S} \frac{P(l_1^m|w_1^n).P(w_1^n)}{P(l_1^m)}$$

As $P(l_1^m)$ is same for all w_1^n , so the denominator does not change the equation, simplifying to the following expression.

$$P(W|L) = \operatorname{argmax}_{w_1^n \in S} P(l_1^m|w_1^n).P(w_1^n)$$

where

$$P(l_1^m|w_1^n) = P(l_1, l_2, l_3, \ldots l_m|w_1^n)$$
$$= P(l_1|w_1^n) * P(l_2|w_1^n l_1) * P(l_3|w_1^n l_1 l_2) * P(l_4|w_1^n l_1 l_2 l_3) * \ldots P(l_m|w_1^n l_1 l_2 l_3 \ldots l_{m-1})$$

Assuming that a ligature l_i depends only on the word sequence w_1^n and its previous ligature l_{i-1}, and not the ligature history, the above equation can be simplifed as follows.

$$P(l_1^m|w_1^n) = P(l_1|w_1^n) * P(l_2|w_1^n l_1) * P(l_3|w_1^n l_2) * P(l_4|w_1^n l_3) * \ldots P(l_m|w_1^n l_{m-1})$$
$$= \prod_1^m P(l_i|w_1^n l_{i-1})$$

Further, if it is assumed that l_i depends on the word in which it appears, not whole word sequence, the equation can be further simplified to the following (as probability of l_i within a word is 1).

$$P(l_1^m|w_1^n) = \prod_1^m P(l_i|l_{i-1})$$

Thus, considering bigrams, $P(W|L) =$

$$\operatorname{argmax}_{w_1^n \in S}\left(\prod_1^m (P(l_i|l_{i-1})\right)\left(\prod_{k=1}^n P(w_k|w_{k-1})\right)$$

This gives the maximum probable word sequence among all the alternative word sequences. The precision of the equation can be taken at bigram or trigram level for both ligature and word, giving the following possibilities. Additionally, normalization is also done to better compare different sequences, as each sequences has different number of words and ligatures per word.

- Ligature trigram and word bigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}l_{i-2})) * (\prod_{k=1}^n P(w_k|w_{k-1}))$$

- Ligature bigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1})) * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))$$

- Ligature trigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}l_{i-2})) * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))$$

- Normalized ligature bigram and word bigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1}))^{1/NW}$$

- Normalized ligature trigram and word bigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}\left((\prod_1^m (P(l_i|l_{i-1}l_{i-2}))^{1/NL}\right) * (\prod_{k=1}^n P(w_k|w_{k-1}))^{1/NW}$$

- Normalized ligature bigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))^{1/NW}$$

- Normalized ligature trigram and word trigram based technique

$$P(W) = \operatorname{argmax}_{w_1^n \in S}(\prod_1^m (P(l_i|l_{i-1}l_{i-2}))^{1/NL} * (\prod_{k=1}^n P(w_k|w_{k-1}w_{k-2}))^{1/NW}$$

In the current work, all the above techniques are used and the best sequence from each one is shortlisted. Then the word sequence which occurs the most times in this shortlist is finally selected.

NL represents the number of ligature bigrams or trigrams and NW represents the number of word bigram or trigrams that exist in the given sentence.

4 Results and Discussion

The model is tested on a corpus of 150 sentences composed of 2156 words and 6075 ligatures. In these sentences, 62 words are unknown, i.e. the words that do not exist in our dictionary. The average length of the sentence is 14 words and 40.5 ligatures. The average length of word is 2.81 ligatures. All the techniques are tested with a beam value, *k*, of 10, 20, 30, 40, and 50.

The results can be viewed from two perspectives: sentence identification rate, and word identification rate. A sentence is considered incorrect even if one word of the sentence is identified wrongly. The technique gives the sentence identification rate of 76% at the beam value of 30. At word level, Normalized Ligature Trigram Word Trigram Technique outperforms other techniques and gives a 96.10% word identification rate at the beam value of 50.

The normalized data gives much better prediction compared to the un-normalized data.

Sentence identification errors depend heavily on the unknown words. For example, at the beam value of 30 we predict 38 incorrect sentences, of which 25 sentence level errors are due to unknown-words and 13 errors are due to known word identification errors. Thus improving system vocabulary will have significant impact on accuracy.

Many of the word errors are caused due to insufficient cleaning of word the larger corpus. Though the words with frequency greater than 50 from the 18 million word corpus have been cleaned, the lower frequency words cause these errors. For example word list still contains "بنیادپر"(*bunyad per*, "depends on"), "سےتقسیم " (*se taqseem*, "divided by") with frequency of 40 and 5 respectively, and each should be two words with a space between them. If low frequency words are also cleaned results will further improve, though it would take a lot of manual effort.

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	110/150	73.33%	2060/2156	95.55%	2024/2092	96.75%	36/64	56.25%
20	112/150	74.67%	2066/2156	95.83%	2027/2092	96.89%	39/64	60.94%
30	114/150	76%	2062/2156	95.64%	2019/2083	96.93%	43/73	58.90%
40	105/150	70%	2037/2156	94.48%	2000/2092	95.60%	37/64	57.81%
50	106/150	70.67%	2040/2156	94.62%	2000/2092	95.60%	40/64	62.50%

Table 3. Results changing beam width *k* of the tree

Technique	Total sentences identified	%age	Total words identified	%age	Total known words Identified	%age	Total unknown words identified	%age
Ligature Bigram	50/150	33.33%	1835/2156	85.11%	1806/2092	86.33%	29/64	45.31%
Ligature Bigram Word Bigram	68/150	45.33%	1900/2156	88.13%	1865/2092	89.15%	35/64	54.69%
Ligature Bigram Word Trigram	83/150	55.33%	1960/2156	90.91%	1924/2092	91.97%	36/64	56.25%
Ligature Trigram	16/150	10.67%	1637/2156	75.93%	1610/2092	76.96%	27/64	42.19%
Ligature Trigram Word Bigram	42/150	28%	1776/2156	82.38%	1746/2092	83.46%	30/64	46.88%
Ligature Trigram Word Trigram	62/150	41.33%	1868/2156	86.64%	1835/2092	87.72%	33/64	51.56%
Normalized Ligature Bigram Word Bigram	90/150	60%	2067/2156	95.87%	2024/2092	96.75%	43/64	67.19%
Normalized Ligature Bigram Word Trigram	100/150	66.67%	2070/2156	96.01%	2028/2092	96.94%	42/64	65.63%
Normalized Ligature Trigram Word Bigram	93/150	62%	2071/2156	96.06%	2030/2092	97.04%	41/64	64.06%
Normalized Ligature Trigram Word Trigram	101/150	67.33%	2072/2156	96.10%	2030/2092	97.04%	42/64	65.63%
Word Bigram	47/150	31.33%	1827/2156	84.74%	1796/2092	85.85%	31/64	48.44%
Word Trigram	74/150	49.33%	1937/2156	89.84%	1903/2092	90.97%	34/64	53.13%

Table 4. Results for all techniques for the beam value of 50

Errors are also caused if an alternate ligature sequence exists. For example the proper noun "کارتک" (*kartak*) is not identifiec as it does not exist in dictionary, but the alternate two word sequence "کار تک" (*kar tak*, "till the car") is valid.

This work uses the knowledge of ligature grams and word grams. It can be further enhanced by using the character grams. We have tried to clean the corpus. Further cleaning and additional corpus will improve the results as well. Improvement can also be achieved by handling abbreviations and English words transliterated in the text. The unknown word detection rate can be increased by applying POS tagging to further help rank the multiple possible sentences.

5 Conclusions

This work presents an initial effort on statistical solution of word segmentation, especially for Urdu OCR systems. This work develops a cleaned corpus of half a million Urdu words for statistical training of ligature based data, which is now available for the research community. In addition, the work develops a statistical model for word segmentation using ligature and word statistics. Using ligature statistics improves upon using just the word statistics. Further normalization has significant impact on accuracy.

References

Aroonmanakun, W. (2002). *Collocation and Thai Word Segmentation.* In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, (pp. 68-75). Pathumthani.

Chang, Jyun-Shen, Chen, S.-D., Zhen, Y., Liu, X.-Z., & Ke, S.-J. (1992). *Large-corpus-based methods for Chinese personal name recognition.* Journal of Chinese Information Processing , 6 (3), 7-15.

Chen, F., & Goodman, T. (1996). *An Empirical Study of Smoothing Techniques for Language Modeling.* In the Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, (pp. 310-318).

Church, K. W., & Gale, W. A. (1991). *A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams*. Computer Speech and Language , 5, 19-54.

Dai, J.-C., & Lee, H.-J. (1994). *Paring with Tag Information in a probabilistic generalized LR parser*. International Conference on Chinese Computing. Singapore.

Durrani, N. (2007). *Typology of word and automatic word Segmentation in Urdu text corpus*. Thesis, National University of Computer & Emerging Sciences, Lahore , Pakistan.

Durrani, N., Hussain, S. (2010). *Urdu Word Segmentation*, In the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), Los Angeles, US.

Haizhou, L., & Baosheng, Y. (1998). *Chinese Word Segmentation*. In Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation, (pp. 212-217).

Haruechaiyasak, C., Kongyoung, S., & Dailey, M. N. (2008). *A Comparative Study on Thai Word Segmentation Approaches*.

Hussain, S. (2008). *Resources for Urdu Language Processing*. In Proceedings of the Sixth Workshop on Asian Language Resources.

Ijaz, M., Hussain, S. (2007). *Corpus Based Urdu Lexicon Development,* in the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan.

Jurafsky, D., & Martin., l. J(2000). *Speech and Language Processing: An Introduction to Natural Language Processing (1st ed.)*. Computational Linguistics and Speech Recognition Prentice Hall.

Khankasikam, K., & Muansuwan, N. (n.d.). Thai Word Segmentation a Lexical Semantic Approach.

Khankasikam, K., & Muansuwan, N. (2005). *Thai Word Segmentation a Lexical Semantic Approach*. In the Proceedings of the Tenth Machine Translation Summit, (pp. 331-338). Thailand.

Lehal, G. (2009). *A Two Stage Word Segmentation System for Handling Space Insertion Problem in Urdu Script*. World Academy of Science, Engineering and Technology 60.

MacKay, D. J., & Peto, L. C. (1995). *A Hierarchical Dirichlet Language Mode*. Natural Language Engineering , 1 (3), 1-19.

Mukund, S. & Srihari, R. (2009). *NE Tagging for Urdu based on Bootstrap POS Learning*. In the Proceedings of CLIAWS3, Third International Cross Lingual Information Access Workshop, pages 61–69, Boulder, Colorado, USA.

Naseem, T., & Hussain, S. (2007). *Spelling Error Trends in Urdu,* In the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan.

Pascale, F., & Dekai, W. (1994). *Statistical augmentation of a Chinese machine readable dictionary*. Proceedings of the Second Annual Workshop on Very Large Corpora, (pp. 69-85).

Poowarawan, Y. (1986). *Dictionary-based Thai Syllable Separation*. Proceedings of the Ninth Electronics Engineering Conference.

Richard Sproat, C. S. (1996). *A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*. Computational Linguistics , 22 (3).

Sajjad, H. (2007). *Statistical Part-of-Speech for Urdu*. MS thesis, National University of Computer and Emerging Sciencies, Centre for Research in Urdu Language Processing, Lahore , Pakistan.

Sornlertlamvanich, V., Potipiti, T., & charoenporn, T. (2000). *Automatic Corpus-Based Thai Word Algorithm Extraction with the C4.5 Learning*. Proceedings of the 18th conference on Computational linguistics.

Sproat, R., Shih, C., Gale, W., & Chang, N. (1996). *A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*. Computational Linguistics , 22 (3).

Theeramunkong, T., & Usanavasin, S. (2001). *Non-Dictionary-Based Thai Word Segmentation Using Decision Trees*. Proceedings of the first international conference on Human language technology research.

Urdu-Nepali-English Parallel Corpus. (n.d.). Retrieved from Center for Research in Urdu Language Processing: *http://www.crulp.org/software/ling_resources/urdunepalienglishparallelcorpus.htm*

Wang, X.-J., Liu, W., & Qin, Y. (2007). A Search-based Chinese Word Segmentation Method. 16th International World Wide Web Conference.

Wong, P.-k., & Chan, C. (1996). *Chinese Word Segmentation based on Maximum Matching and Word Binding Force*. In Proceedings of the 16th conference on Computational linguistics.